Guida per l'utente di Wine

Traduzione in italiano di Andrea Montagner

dgumo@tiscali.it

(v. 1.0 del 25 ottobre 2008)

Tavola dei contenuti

Capitolo 1. Introduzione

Sommario/Presentazione

Scopo di questo documento e pubblico a cui è diretto

Questo documento, chiamato "Guida per l'utente di Wine", è sia una semplice guida per l'installazione, sia una ampia guida di riferimento. Questa guida è destinata sia ai nuovi utenti di Wine, sia a quelli esperti, offrendo istruzioni complete passo-passo per l'installazione e la configurazione, come pure caratterizzando il vasto materiale di riferimento con documentazione su tutte le caratteristiche della configurazione e sulle aree di supporto.

Domande e commenti ulteriori

Se, dopo aver esaminato questa guida, le FAQ ed altra documentazione di rilievo, c'è ancora qualcosa che non riuscite a comprendere, noi ameremmo sentirlo da voi. La sezione delle liste postali (http://www.winehq.oeg/site/forums) contiene diverse liste postali ed un canale IRC, tutti validi posti per cercare aiuto e offrire suggerimenti. Se siete particolarmente abili e credete che qualcosa sia da spiegare meglio, potete preparare una segnalazione di errore (http://bugs.winehq.org/) o inviare una correzione (http;//www.winehq.org/site/sendiing_patches) alla documentazione stessa di Wine.

Sommario dei contenuti/Fasi da percorrere

Per essere in grado di utilizzare Wine, dovete prima avere una installazione funzionante. Questa guida vi aiuterà a trasformare il vostro sistema dal vuoto privo di Wine ad uno che vanta una nuova ed aggiornata installazione di Wine. La prima fase, “Procurarsi Wine”, illustra i vari metodi per collocare i file di Wine nel vostro computer. La seconda fase, “Configurare Wine”, mostra come personalizzare una installazione di Wine in base alle vostre individuali necessità. La fase finale, “Far funzionare Wine”, tratta degli specifici passi da intraprendere per ottenere che una particolare applicazione giri meglio sotto Wine, e fornisce utili collegamenti in caso abbiate bisogno di ulteriore aiuto.

Avvio rapido

La procedura di installazione e partenza di Wine può essere così riassunta:

- Procuratevi una distribuzione come indicato in “Procurarsi Wine” e guardate la pagina Wine Downloads (http://www.winehq.org/site/download). Per l'utente saltuario o novizio la cosa più semplice è prendere una distribuzione in rpm.
- Configurate wine usando il comando *`winecfg`*. Per la maggioranza della gente ora wine dovrebbe essere utilizzabile.
- Per provare la vostra installazione, avviate il gestore di file simile a Windows 3.1 di wine utilizzando il comando `wine winefile`.
- Avviate wine usando il comando `wine filespec/nomeapp.exe`. Il primo

comando da impartire sarà quello per installare un pacchetto, normalmente qualcosa come `wine /media/cdrom/setup.exe` o equivalente.

Cos'è Wine?

Windows e Linux

Differenti programmi software sono progettati per sistemi operativi differenti e la maggioranza non funziona nei sistemi per cui non sono stati ideati. I programmi Windows, ad esempio, non gireranno sotto Linux perché contengono istruzioni che il sistema non comprende sino a che non vengono tradotte dall'ambiente Windows. Allo stesso modo i programmi Linux non gireranno sotto il sistema operativo Windows perché quest'ultimo è incapace di interpretare tutto delle loro istruzioni.

Tale situazione rappresenta un problema fondamentale per chiunque voglia far girare software sia su Windows, sia su Linux. Una comune soluzione di questo problema è quella di installare entrambi i sistemi operativi sullo stesso computer, il cosiddetto "dual boot" o "doppio avvio". Quando serve un programma Windows, l'utente avvia la macchina in Windows per farlo funzionare: quando invece serve un programma Linux, l'utente allora riavvia la macchina in Linux. Tale opzione presenta una grossa difficoltà: l'utente non solo deve sopportare la frustrazione di frequenti riavvi, ma anche i programmi di ambedue le piattaforme non possono girare in contemporanea. Avere Windows in un sistema crea un onere ulteriore: il software è costoso, richiede una separata partizione del disco ed è incapace di leggere la maggior parte dei formati di filesystem, rendendo difficoltoso lo scambio dei dati tra sistemi operativi.

Cos'è Wine e come può aiutarmi?

Wine rende possibile il funzionamento di programmi Windows in qualsiasi sistema operativo simil-Unix, in modo particolare in Linux. Nel suo cuore, Wine è una trasposizione della libreria Windows Application Programing Interface (API), che agisce da ponte tra il programma Windows e Linux. Pensate a Wine come ad uno strato di compatibilità quando un programma Windows tenta di eseguire una funzione che normalmente Linux non comprende. Wine tradurrà l'istruzione di quel programma in una supportata dal sistema. Per esempio, se un programma chiede al sistema di creare un *pushbutton* Windows o un campo di inserimento testo, Wine convertirà quella istruzione nel suo equivalente Linux in forma di comando al gestore di finestre che utilizza il protocollo standard X11.

Se avete accesso al codice sorgente del programma Windows, Wine può essere anche utilizzato per ricompilare il programma in un formato che Linux possa comprendere con maggiore facilità. Wine ha ancora bisogno di lanciare il programma nella sua forma ricompilata, tuttavia sussistono molti vantaggi nel compilare un programma Windows nativamente con Linux. Per maggiori informazioni, consultate la "Winelib User Guide".

Caratteristiche di Wine

Durante il corso del suo sviluppo Wine è cresciuto costantemente nelle caratteristiche che supporta e nei programmi che può far funzionare. Segue un elenco parziale di tali caratteristiche:

- Supporto per il funzionamento dei programmi Win32 (Win 95/98, NT/2000/XP), Win16 (Win 3.1) e DOS
- Utilizzo facoltativo di file DLL di venditori esterni (come quelli inclusi con Windows)
- Visualizzazione grafica basata su X11 che consente la rappresentazione remota su qualsiasi terminale X, così come in modalità "console"
- Desktop in un riquadro o finestre mescolabili
- Supporto DirectX per i giochi
- Buon supporto per svariate schede sonore comprensivo di OSS e ALSA
- Supporto per periferiche d'ingresso alternative
- Stampa: gestore d'interfaccia PostScript (psdrv o PostScript interface Driver) per i servizi standard di stampa Unix PostScript
- Supporto per modem e unità seriali
- Supporto per la rete Winsock TCP/IP
- Supporto interfaccia ASPI (SCSI) per scanner, masterizzatori CD ed altre periferiche
- Supporto avanzato per unicode e linguaggi stranieri
- Correttore (debugger) Wine completo e messaggi configurabili di registrazione dei tracciamenti per una individuazione più semplice degli errori

Versioni di Wine

Wine di Wine HQ

Wine è un progetto a sorgente aperto e conseguentemente esistono molte differenti versioni di Wine tra cui potete scegliere. La versione standard di Wine arriva con rilasci intermittenti (circa due volte al mese) e può essere scaricata da internet sia in formato binario preimpacchettato, sia in formato pronto per la compilazione del codice sorgente. In alternativa potete installare una versione di sviluppo di Wine utilizzando il più recente codice sorgente disponibile sul repository Git. Guardate il prossimo capitolo, “Procurarsi Wine”, per maggiori dettagli.

Altre versioni di Wine

Esistono numerosi programmi derivati dal codice base standard di Wine in un modo o in un altro. Alcuni di questi sono prodotti commerciali di società che contribuiscono attivamente al progetto Wine.

Tali prodotti tentano di elevarsi o distinguersi dalla versione normale di Wine offrendo una maggiore compatibilità, una configurazione più semplice e supporto commerciale. Se cercate tali cose, è una buona idea valutare l'acquisto di questi prodotti.

Tabella 1-1. Offerte varie di Wine

Prodotto	Descrizione	Forma di distribuzione

CodeWeavers CrossOver Office (http://www.codeweavers.com/products/office)	CrossOver Office vi consente di installare la vostra preferita applicazione Windows di produttività sotto Linux, senza la necessità di una licenza del sistema operativo Microsoft. CrossOver comprende una interfaccia a singolo clic semplice da usare che rende facile e rapida l'installazione di un'applicazione Windows.	Commerciale: disponibile dimostrativo pienamente funzionante per 30 giorni.
CodeWeavers CrossOver Office Server Edition. (http://www.codeweavers.com/products/cxofficeserver)	CodeWeavers CrossOver Office Server Edition vi consente di installare la vostra preferita applicazione Windows di produttività in un ambiente distribuito di thin.client sotto Linux, senza la necessità di licenze del sistema operativo Windows per ciascuna macchina cliente. CrossOver Office Server Edition vi permette letteralmente di soddisfare i bisogni di centinaia di utenti in contemporanea, tutti da un unico server.	

Alternative a Wine che potreste prendere in considerazione

Ci sono molte maniere per far funzionare del software oltre che con Wine. Se state considerando l'impiego di Wine per far girare un'applicazione, potreste desiderare di valutare la percorribilità di questi approcci nel caso incontriate difficoltà.

Applicazioni native

Invece di far funzionare una particolare applicazione Windows con Wine, un'alternativa frequentemente percorribile è semplicemente quella di lanciare un'applicazione differente. Molte applicazioni Windows, soprattutto quelle usate più comunemente come i riproduttori multimediali, le messaggerie istantanee e i programmi di scambio file, hanno degli equivalenti a codice aperto piuttosto validi. Inoltre un cospicuo numero di programmi Windows è stato portato direttamente in Linux, eliminando interamente la necessità di Wine (o Windows).

Un altro sistema operativo

Probabilmente il metodo più ovvio per fare in modo che un'applicazione Windows funzioni è quello di avviarla sotto Windows. Tuttavia sicurezza, costi delle licenze, retrocompatibilità e questioni di efficienza delle macchine possono rendere questo un arduo proponimento: il che spiega perché Wine è in primo luogo così utile.

Un'altra alternativa è l'utilizzo di ReactOS (http://www.reactos.com), che è un'alternativa di Windows completamente a codice aperto. ReactOS condivide pesantemente il codice con il progetto Wine, ma, piuttosto di far girare le applicazioni Windows sotto Linux, le fa funzionare con il kernel di ReactOS. ReactOS offre anche la compatibilità con i file dei driver Windows, consentendo l'uso di hardware senza il funzionamento di driver Linux.

Macchine virtuali

Piuttosto di installare nella vostra macchina un sistema operativo completamente nuovo, potete invece far funzionare una macchina virtuale a livello software ed installare su di essa un sistema operativo differente. Così potreste eseguire un sistema Linux e contemporaneamente far funzionare Windows con la sua applicazione in una macchina virtuale nello stesso hardware. Le macchine virtuali vi permettono di installare e far girare non solo tipi differenti di Windows sullo stesso macchinario, ma anche altri sistemi operativi, compreso ReactOS.

Esistono qui parecchie offerte differenti di macchine virtuali ed alcune sono pure in grado di emulare un'apparecchiatura X86 su piattaforme diverse. Bochs (http://bochs.sourceforge.net/) e QEMU (http://www.quemu.org/) a sorgenti aperti sono capaci di far funzionare sia Windows che ReactOS in modo virtuale. Altre offerte di macchine virtuali commerciali includono VMware (http;//www.vmware.com/) e VirtualPC di Microsoft (http://www.microsoft.com/windows/virtualpc/).

Tuttavia ci sono dei rilevanti inconvenienti nell'uso delle macchine virtuali. Diversamente da Wine, questi programmi *sono* degli emulatori, cosicché sussiste una inevitabile perdita di velocità che può essere abbastanza consistente. Inoltre, il funzionamento di un'applicazione all'interno di una macchina virtuale impedisce la piena integrazione dell'applicazione con l'ambiente ospitante. Per esempio, non sareste in grado di avere le icone della barra di sistema di Windows oppure i collegamenti dei programmi posizionati tra quelli del vostro desktop Linux, dal momento che invece le applicazioni Windows devono risiedere completamente nella macchina virtuale.

Capitolo 2. Procurarsi Wine

Metodi per installare Wine

Una volta deciso che Wine va bene per le vostre necessità, il passo successivo è quello di stabilire come lo volete installare. Esistono tre metodi per installare Wine da WineHQ, ognuno con i propri vantaggi e svantaggi.

Installazione da un pacchetto

Naturalmente il metodo più semplice per installare Wine è quello di utilizzare una versione preimpacchettata di Wine. Questi pacchetti contengono i file binari di Wine già pronti per funzionare, compilati specificatamente per la vostra distribuzione, e vengono controllati regolarmente dagli impacchettatori sia nel funzionamento che nella completezza.

I pacchetti sono il metodo raccomandato per l'installazione di Wine. Li rendiamo facilmente disponibili nella pagina degli scaricamenti di WineHQ (http://www.winehq.org/site/download) e sono sempre gli ultimi pacchetti disponibili. Essendo popolari, i pacchetti di Wine si possono trovare anche altrove nei depositi (*repository*) ufficiali delle distribuzioni. Questi, tuttavia, qualche volta possono essere sorpassati, a seconda della distribuzione. I pacchetti si possono aggiornare con facilità e molte distribuzioni possono farlo senza problemi in pochi clic. E' anche possibile realizzare il vostro pacchetto binario installabile partendo da un pacchetto di sorgenti, sebbene ciò vada oltre gli scopi di questa guida.

Installazione da un archivio di sorgenti.

Talvolta i pacchetti di Wine non soddisfano esattamente le vostre necessità. Forse non sono disponibili per la Vostra architettura o distribuzione, o forse volete costruire Wine utilizzando ottimizzazioni del vostro compilatore o con alcune opzioni disabilitate, oppure forse avete bisogno di modificare una particolare parte del codice sorgente prima della compilazione. Trattandosi di un progetto a sorgente aperto, voi avete la possibilità di fare tutte queste cose con il codice sorgente di Wine (che viene fornito con ogni rilascio di Wine). Tale metodo di installazione può essere effettuato scaricando un archivio dei sorgenti di Wine e compilando da linea di comando. Se vi trovate a vostro agio con questo genere di cose e avete delle esigenze particolari, questa opzione può essere per voi.

Procurarsi gli archivi di Wine è semplice. Ad ogni rilascio, mettiamo un pacchetto di sorgenti in formato compresso tar.bz2 nella pagina degli scaricamenti di WineHQ (http://www.winehq.org/site/download). Compilare e installare Wine dai sorgenti è leggermente più difficoltoso rispetto all'uso di un pacchetto, tuttavia tratteremo ciò in profondità e tenteremo di tenervi per mano lungo questo percorso.

Installazione da un albero di Git

Se desiderate provare lo sviluppo di Wine tecnologicamente più avanzato oppure vi piacerebbe voi stessi aiutare a sviluppare Wine, potreste scaricare il codice sorgente di Wine più recente in assoluto dal nostro deposito (*repository*) Git. Le istruzioni per lo scaricamento dal deposito Git

di Wine sono disponibili su http://www.winehq.org/site/git.

Per favore, annotatevi che restano ancora validi gli abituali avvisi circa l'uso di una versione in corso di sviluppo. Il codice sorgente nel deposito Git è ampiamente da collaudare e potrebbe anche non compilarsi correttamente. E' tuttavia il modo migliore per verificare come Wine funzionerà con la prossima versione e, se state modificando il codice sorgente, è meglio che vi procuriate la copia più recente. Il deposito Git è anche utile per i manutentori delle applicazioni interessati a provare se un'applicazione continuerà a funzionare bene con il prossimo rilascio o se una recente pezza migliori le cose. Se vi interessa aiutarci a far funzionare un'applicazione con Wine, guardate l'HowTo (http://www.winehq.org/site/howto).

Installare Wine da un pacchetto

Installare da zero un pacchetto

L'installazione di un pacchetto su un sistema nuovo è estremamente semplice. Semplicemente scaricate ed installate il pacchetto utilizzando qualsiasi programma di utilità vi fornisca la vostra distribuzione. Solitamente non c'è bisogno di rimuovere espressamente vecchi pacchetti prima dell'installazione, dal momento che le moderne distribuzioni Linux dovrebbero aggiornarli ed installarli automaticamente. Tuttavia, se avete installato Wine da codice sorgente, dovreste rimuoverlo prima di installare un pacchetto Wine. Guardate la sezione sulla disinstalllazione di Wine dal sorgente per le giuste istruzioni.

Distribuzioni differenti

Wine funziona su una cospicua quantità di differenti distribuzioni Linux, così come su altri sistemi simil-Unix come Solaris e FreeBSD, ciascuno con il proprio specifico modo di installare e gestire i pacchetti. Tuttavia, fortunatamente, a tutti loro si applicano le stesse idee generali e l'installazione di Wine non dovrebbe essere più difficile rispetto a qualsiasi altro software, non importa quale distribuzione usiate. La disinstallazione è ugualmente facile e nelle moderne distribuzioni Linux viene effettuata attraverso la medesima semplice interfaccia dell'installazione del pacchetto.

In questa guida non vogliamo trattare nello specifico l'installazione e la disinstallazione dei pacchetti di Wine con i vari metodi dei sistemi di gestione dell'impacchettamento e dello spacchettamento, tuttavia note aggiornate di installazione per distribuzioni particolari si possono trovare sul sito web di WineHQ in HowTo (http://www.winehq.org/site/howto). Se vi serve ulteriore aiuto per vedere quanto semplice è l'installazione di un pacchetto di Wine, suggeriamo di consultare la documentazione della vostra distribuzione, i fori di supporto o i canali IRC.

Installare Wine da sorgente

Prima di installare Wine da sorgente, assicuratevi di aver disinstallato tutti i pacchetti binari di Wine che avreste potuto avere nel vostro sistema. L'installazione da sorgente richiede l'uso della finestra di terminale così come una copia integrale del codice sorgente di Wine. Una volta scaricato il sorgente da Git o estrattolo da un archivio, entrate nella sua directory e seguite i passi

rimanenti.

Procurarsi le dipendenze per la creazione

Wine utilizza molte librerie a sorgente aperto durante il suo funzionamento. Sebbene non dipenda rigorosamente da tali librerie, molte sue funzionalità traggono benefici nell'averle a disposizione al momento della compilazione. Tempo addietro molti problemi degli utenti erano causati da gente che non aveva le necessarie librerie di sviluppo al momento della creazione di Wine dal sorgente: per questo ed altri motivi raccomandiamo di installare tramite pacchetti binari o pacchetti di sorgenti per la creazione che possano soddisfare automaticamente le loro dipendenze di creazione.

Se preferite installare a mano le dipendenze di creazione, esistono diverse maniere per vedere se state tralasciando alcune utili librerie di sviluppo. L'approccio più diretto è quello di osservare i dati in uscita del programma *configure* prima di compilare Wine e di vedere se si è perso qualcosa di importante: se così fosse, installate semplicemente ciò che è stato trascurato e riavviate *configure* prima della compilazione. Potete anche controllare il file generato da *configure* (include/config.h) e vedere quali file *configure* cerca ma non trova.

Compilazione di Wine

Una volta installate le necessarie dipendenze di creazione, siete pronti per compilare il pacchetto. Nella finestra di terminale, dopo aver raggiunto l'albero dei sorgenti di Wine, lanciate i seguenti comandi:

```
$ ./configure
$ make depend
$ make
# make install
```

L'ultimo comando richiede i privilegi di root. Sebbene non dovreste mai avviare Wine in qualità di root, avrete bisogno di installarlo in questo modo.

Disinstallazione di Wine da sorgente

Per disinstallare Wine da sorgente, navigate ancora una volta fino alla stessa cartella dei sorgenti che avete usato per l'installazione con il terminale. Poi lanciate il seguente comando:

```
# make uninstall
```

Questo comando richiederà i privilegi di root e dovrebbe rimuovere tutti i file binari di Wine dal vostro sistema. La vostra configurazione di Wine e le applicazioni collocate nella vostra directory personale non verranno, comunque, rimosse in modo che siete liberi di installare una nuova versione di Wine oppure di cancellare manualmente tale configurazione.

Capitolo 3. Configurare Wine

La maggior parte delle più comuni modifiche alla configurazione possono essere realizzate con lo strumento Winecfg. Attraverseremo una semplice introduzione passo-passo a Winecfg ed evidenzieremo le opzioni disponibili. Nella successiva sezione esamineremo modifiche maggiormente sofisticate che potete apportare usando regedit così come fornendo una completa guida di riferimento a tutte le impostazioni di configurazione di Wine. Infine alcune cose che potreste voler configurare ricadono al di fuori degli scopi di Winecfg e regedit e perciò le sorvoleremo.

Uso di Winecfg

In passato Wine utilizzava uno speciale file di configurazione che poteva essere trovato in `~/.wine/config`. Se state ancora utilizzando una vecchia versione di Wine che fa riferimento ancora a quel file (precedente al giugno 2005), dovreste aggiornarvi prima di procedere oltre. Tutte le impostazioni vengono ora conservate dentro il registro (*registry*) e vengono consultate alla partenza di Wine.

Winecfg dovrebbe essere stato installato nel vostro computer insieme al resto dei programmi di Wine. Se non sapete come avviarlo, provate a lanciare il comando: `$` **`/usr/local/bin/winecfg`** o eventualmente solo: `$` **`winecfg`**

All'avvio del programma, noterete che ci sono delle schede in cima alla finestra per:

·Applications (Applicazioni)

·Libraries (Librerie)

·Graphics (Grafica)

·Desktop Integration (Integrazione desktop)

·Drives (Drive)

·Audio (Audio)

·About (About)

Le modifiche alle impostazioni delle schede *Application* e *Libraries* avranno il maggior effetto nel rendere funzionante un'applicazione. Le altre impostazioni sono mirate ad ottenere che Wine stesso si comporti come voi volete.

Nota: Le schede Application, Libraries e Graphics sono collegate assieme! Se avete Default Settings selezionato in Application, tutte le modifiche effettuate con le schede Libraries e Graphics verranno cambiate per tutte le applicazioni. Se avete configurato una specifica applicazione nella scheda Application e l'avete selezionata, allora tutte le modifiche fatte in Libraries o Graphics avranno effetto solo per tale applicazione. Ciò consente la personalizzazione per specifiche applicazioni.

Impostazioni di Application

Wine ha la capacità di imitare il comportamento di differenti versioni di Windows. In generale la

maggior differenza è se Wine si comporta da versione Win9x o NT. Alcune applicazioni richiedono uno specifico comportamento per funzionare e la modifica di tale impostazione potrebbe determinare l'utilizzo di una applicazione difettosa. Recentemente la versione Windows predefinita di Wine è diventata Windows 2000. E' noto che molte applicazioni saranno più performanti se sceglierete Windows 98.

Avrete notato che nella scheda c'è una voce *Default Settings* [Impostazioni predefinite]. Se selezionerete quella,vedrete la corrente *Windows Version* prestabilita per tutte le applicazioni. E' meglio che un'applicazione problematica venga configurata separatamente dalle impostazioni predefinite. Per fare ciò:

·1 Cliccate sul bottone *Add application*

·2 Scorrete finché trovate l'.exe

·3 Dopo che è stata aggiunta, potete scegliere la specifica versione di Windows che Wine emulerà per tale applicazione.

Impostazioni delle Librerie

Allo stesso modo alcune applicazioni richiedono delle specifiche librerie per funzionare. Wine riproduce le librerie di sistema Windows (le cosiddette DLL native) con versioni completamente personalizzate, progettate per funzionare esattamente nello stesso modo ma senza richiedere licenze da Windows. Wine ha molte lacune note nelle sue versioni incorporate, ma in molti casi la funzionalità è sufficiente. L'uso delle DLL incorporate assicura che il vostro sistema sia Microsoft-esente. Tuttavia Wine possiede la capacità di caricare le native DLL di Windows.

DLL Overrides

Non è sempre possibile avviare un'applicazione con le DLL incorporate. Talvolta le DLL native semplicemente funzionano meglio. Dopo che avrete individuato una DLL nativa in un sistema Windows, dovrete metterla nel posto adatto affinché Wine possa trovarla e poi configurarla in modo da poterla utilizzare. In genere il posto in cui dovete collocarla è la directory che avete configurato in modo da essere `c:\windows\system32` (maggiori informazioni su ciò nella sezione drives). Esistono quattro DLL che non dovreste mai provare, le versioni native di: `kernel32.dll`, `gdi32.dll`, `user32.dll` e `ntdll.dll`. Queste librerie richiedono l'accesso a basso livello al kernel di Windows che semplicemente non esiste con Wine.

Con ciò in mente, una volta copiata la DLL, dovete soltanto dire a Wine di provare ad usarla. Potete configurare Wine affinché scelga a due livelli differenti tra le DLL native e quelle incorporate. Se avete selezionato *Default Settings* nella scheda *Applications*, le modifiche che apporterete avranno effetto su tutte le applicazioni. Oppure potete scavalcare le impostazioni globali a livello di singola applicazione aggiungendone e selezionandone una nella scheda *Applications*.

Per aggiungere un'eccezione per FOO.DLL, inserite "FOO" nella casella etichettata *New override for library:* e cliccate sul bottone *Add*. Per modificare il comportamento della DLL, selezionatela nella casella *Existing overrides:* e scegliete *Edit*. Normalmente il nuovo ordine di caricamento sarà le librerie Windows native prima di quelle incorporate in Wine (*Native then Builtin*). Potete scegliere anche *native only* (solo native), *builtin only* (solo incorporate) oppure

disabilitarle tutte quante.

Note sulle DLL di sistema

Il gruppo di Wine ha deciso che è necessario creare dei file di DLL fittizi per ingannare molti programmi che controllano l'esistenza di file per stabilire se sia disponibile una particolare funzionalità (come Winsock e il suo collegamento in rete TCP/IP). Se ciò rappresenta un problema per voi, potete creare dei file vuoti nella configurata directory c:\windows\system32 per fare in modo che il programma creda che siano lì e le DLL incorporate in Wine saranno caricate quando il programma le ricercherà veramente (sfortunatamente tools/wineinstall non crea da solo tali file vuoti).

Alle volte le applicazioni cercano anche di ispezionare le risorse di versione dai file fisici (per esempio per stabilire la versione di DirectX). I file vuoti non lo fanno in questo caso: è piuttosto necessario installare dei file con risorse di versione complete. Tale problema è già stato risolto per molti file. Per altri potreste avere ancora bisogno di procurarvi alcuni veri file di DLL per ingannare tali applicazioni con questi.

Esistono naturalmente delle DLL che Wine ora non riesce a riprodurre molto bene (o del tutto). Se non avete un Windows originale da cui copiare le DLL necessarie, potete sempre prenderne qualcuna da siti d'archivio delle DLL Windows che si possono trovare tramite un motore di internet. Per favore accertatevi di rispettare qualsiasi licenza delle DLL che prendete: alcune sono redistribuibili, altre no.

DLL mancanti

Nel caso Wine lamenti l'assenza di una DLL, potreste verificare se tale file è una DLL pubblicamente disponibile oppure una peculiare DLL appartenente al vostro programma (cercando con il suo nome su internet). Dopo aver individuato la DLL, dovete assicurarvi che Wine sia capace di utilizzarla. Di solito le DLL vengono caricate nell'ordine seguente:

1. La directory da cui è stato avviato il programma;
2. La directory attuale;
3. La directory Windows di sistema;
4. La directory Windows;
5. Le directory della variabile PATH.

In breve: sia mettendo le DLL richieste nella vostra directory del programma (potrebbe essere complicato) oppure mettendole nella directory di sistema di Windows. Pure, se possibile, non dovreste probabilmente usare le DLL native basate su NT, poiché il supporto di Wine alle API NT è per qualche aspetto inferiore rispetto a quello relativo alle API Win9x (forse conducendo con le DLL NT anche ad una peggiore compatibilità rispetto ad una configurazione senza finestre!).

Impostazioni grafiche

Di base esistono cinque impostazioni della grafica che potete configurare. Per la maggior parte della gente quelle predefinite vanno bene.

Le poche prime impostazioni influenzano principalmente i giochi e sono in qualche modo autoesplicative. Potete impedire al mouse di abbandonare la finestra di un programma DirectX (cioè un gioco) e l'impostazione predefinita è avere la casella selezionata. Sussistono numerose ragioni per voler fare ciò, non ultima tra queste che è più semplice giocare se il cursore viene confinato in un'area più ristretta. L'altro motivo per attivare tale opzione è per un controllo più preciso del mouse - Wine adatta la posizione del mouse per imitare il modo in cui funziona Windows. Similmente "desktop double buffering" permette aggiornamenti più precisi dello schermo (da cui possono trarre benefici i giochi) e l'impostazione predefinita è quella di lasciarlo attivo. La contropartita è un incremento dell'uso di memoria.

Potreste trovarlo utile per *Emulate a virtual desktop*. In questo caso tutti i programmi gireranno in una finestra separata. Potreste trovarlo utile come modo per testare giochi "bacati" che cambiano (eventualmente non riuscendovi) la risoluzione dello schermo: confinandoli in una finestra si possono effettuare maggiori controlli su di loro forse al prezzo di una diminuita usabilità. Le dimensioni che potreste voler provare sono 640x480 (predefinita) oppure 800x600.

Infine, potete configurare qualche impostazione delle DirectX. Per la maggior parte queste impostazioni vengono rilevate automaticamente, ma potete forzarle a comportarsi in un modo specifico. Alcuni giochi tentano di esplorare il sistema di fondo per vedere se supporta delle particolari funzionalità. Disattivando queste impostazioni Wine non mostrerà la capacità di riprodurre i giochi in una certa maniera. Ciò potrebbe condurre il gioco a funzionare più velocemente a scapito della qualità della grafica, oppure al totale mancato funzionamento dello stesso.

Impostazioni dei drive

Windows richiede una configurazione dei drive piuttosto rigida che Wine imita. La maggior parte delle persone ha familiarità con la notazione standard di "A:", che rappresenta il disco floppy, del drive "C:"che rappresenta il disco primario di sistema, ecc. Wine utilizza il medesimo concetto e mappa quei drive nel sottostante filesystem nativo.

La configurazione dei drive in Wine è relativamente semplice. In Winecfg sotto la scheda *Drives* vedrete dei bottoni per aggiungere e rimuovere i drive disponibili. Quando scegliete di aggiungere un drive, una nuova voce verrà inserita ed apparirà una mappatura predefinita dei drive. Potete modificare dove questi drive puntano cambiando quello che c'è nella casella *Path:*. Se non siete sicuri dell'esatto percorso, potete scegliere "Browse" per ricercarlo. La rimozione di un drive è semplice con la scelta del drive e la pressione di "Remove".

Winecfg possiede la capacità di scoprire automaticamente i drive disponibili nel vostro sistema. Si raccomanda di provare ciò prima di tentare la configurazione manuale dei drive. Battete semplicemente il bottone *Autodetect* per fare in modo che Wine ricerchi i drive nel vostro sistema.

Potreste essere interessati a configurare le impostazioni dei drive al di fuori di winecfg, nel qual caso sareste abbastanza fortunati perché è abbastanza facile. Tutte le impostazioni dei drive risiedono in una speciale directory, `~/.wine/dosdevices`. Ogni "drive" è semplicemente un collegamento a dove si trova realmente. Wine automaticamente predispone due drive la prima volta che lo avviate:

```
$ ls -la ~/.wine/dosdevices/
```

```
lrwxrwxrwx 1 wineuser wineuser 10 Jul 23 15:12 c: -> ../drive_c
lrwxrwxrwx 1 wineuser wineuser  1 Jul 23 15:12 z: -> /
```

Per aggiungere un altro drive, per esempio il vostro CD-ROM, create soltanto un nuovo collegamento che punti ad esso: `$ ln -s /mnt/cdrom ~/.wine/dosdevices/d:`. Prendete nota delle convenzioni di denominazione in stile DOS utilizzate per i collegamenti - il formato è una lettera seguita da due punti, come in "a:". In tal modo, se il collegamento al vostro drive c: punta a ~/.wine/drive_c, potete interpretare i i riferimenti a `c:\windows\system32` nel senso di `~/.wine/drive_c/windows/system32`.

Impostazioni audio

Wine può funzionare con parecchi differenti sottosistemi audio che potete scegliere nella scheda "Audio". winecfg capisce tutti i driver al posto vostro, ma potete selezionare manualmente quale driver verrà usato. Le distribuzioni Linux più datate che utilizzano il kernel 2.4 o precedente normalmente adoperano il driver "OSS". Di solito i kernel 2.6 sono passati ad "ALSA". Il driver "aRts" è stato recentemente disattivato a causa della generale mancanza di manutenzione del sottosistema di "aRts". Se state utilizzando GNOME probabilmente potete usare EsounD. I driver audio OSS e ALSA sono i più collaudati, cosicché si raccomanda di adottarli se possibile. Se dovete usare "Jack", "NAS" o "CoreAudio", già ne sapete la ragione.

Le impostazioni DirectSound sono impiegate principalmente per i giochi. Potete scegliere il livello di accelerazione che vi aggrada, però per la maggioranza della gente "Full" va bene.

Integrazione con il desktop

Wine può caricare temi di Windows, se ne avete a disposizione. Mentre questo non è di certo necessario per l'uso di Wine o delle applicazioni, ciò vi permette di personalizzare l'aspetto e le impressioni di un programma. Wine supporta i più recenti tipi MSStyles di temi. Diversamente dai più vecchi temi di stile Microsoft Plus! Il motore uxtheme supporta i file speciali .msstyles che possono sostituire ii temi di tutti i controlli Windows. Questo è più o meno lo stesso genere di realizzazione dei temi che i moderni desktop Linux hanno supportato per anni. Se vi piace, provate questo:

1. Scaricate un tema Windows. Assicuratevi che contenga un file .msstyles
2. Create un insieme di nuove directory nel vostro drive fittizio Windows: `$` **`mkdir -p ~/.wine/drive_c/windows/Resources/themes/nome-del-vostro-tema`**
3. Spostate il .msstyles nella nuova directory nome-del-vostro-tema.
4. Utilizzate la scheda Desktop Integration di winecfg per selezionare il nuovo tema.

Uso del Registry e di Regedit

Tutte le impostazioni che modificate in Winecfg, ad eccezione di quelle dei drive, vengono in definitiva archiviate nel registro. In Windows questo è un deposito centralizzato delle configurazioni delle applicazioni e del sistema operativo. Allo stesso modo Wine contiene un

registro ed alcune impostazioni non riscontrabili in Winecfg possono essere modificate in questo (in realtà esiste più di una possibilità se avrete bisogno di tuffarvi nel registro per modificare le impostazioni di una applicazione oltre a Wine stesso).

Ora, il fatto che lo stesso Wine usi il registro per conservare le impostazioni è stato controverso. Alcune persone argomentano che questo è troppo simile a Windows. Per ribattere a ciò, ci sono diverse cose da valutare. Per prima cosa, è impossibile evitare di predisporre un registro semplicemente perché le applicazioni se lo aspettano per essere in grado di salvare lì le loro impostazioni. Per fare in modo che Wine registri ed acceda alle impostazioni in un file di configurazione separato sarebbe richiesto un insieme separato di codice per fare sostanzialmente la stessa cosa delle API Win32 che Wine implementa già. Infine, diversamente da Windows, il registro di Wine è scritto in puro testo e può essere modificato usando il vostro editor di testi preferito. Mentre la maggior parte degli amministratori di sistema (e degli sviluppatori di Wine) sani di mente impazziscono per la natura contorta del registro di Windows, è ancora necessario per Wine supportarlo in qualche modo.

Struttura del registro

Orbene, detto ciò, scaviamo un po' nel registro per vedere come è confezionato. Il registro di Windows è una elaborata struttura ad albero e nemmeno la maggior parte dei programmatori Windows sono a conoscenza di come è costituito il registro, con i suoi diversi "assembramenti" e i numerosi collegamenti tra di essi: una trattazione completa è fuori dello scopo di questo documento. Tuttavia qui ci sono le chiavi di registro basilari di cui potreste per ora aver bisogno di conoscere:

HKEY_LOCAL_MACHINE

Questa fondamentale chiave di base (in Windows 9x viene conservata nel file nascosto `system.dat`) contiene tutto ciò che riguarda la corrente installazione di Windows. Viene spesso abbreviata in HKLM.

HKEY_USERS

Questa fondamentale chiave di base (in Windows 9x viene conservata nel file nascosto `user.dat`) contiene i dati della configurazione di ogni utente dell'installazione.

HKEY_CLASSES_ROOT

Questa è un collegamento a HKEY_LOCAL_MACHINE\SoftwareClasses. Contiene i dati che descrivono cose come le associazioni dei file, i gestori dei documenti OLE e le classi COM.

HKEY_CURRENT_USER

Questa è un collegamento a HKEY_USERS\vostro_nome, cioè la vostra configurazione personale.

I file di registro

Ora, ciò di cui vi state meravigliando è come avvenga la traduzione nella struttura di Wine. Il formato del registro sopradescritto in realtà vive in tre file differenti contenuti nella directory ~/.wine di ogni utente:

`system.reg`

Questo file contiene HKEY_LOCAL_MACHINE.

`user.reg`

Questo file contiene HKEY_CURRENT_USER.

`userdef.reg`

Questo file contiene HKEY_USERS\Default (cioè le impostazioni predefinite degli utenti).

Questi file vengono creati automaticamente da **wineprefixcreate** la prima volta che usate Wine. Un insieme di impostazioni globali viene riposto in `c:\windows\inf\wine.inf` e viene elaborato dal programma `rundll32.exe`. La prima volta che fate funzionare Wine il file `wine.inf` viene processato per riempire il registro iniziale. Per maggiori dettagli controllate lo script `wineprefixcreate` onde vedere com'è realizzato il tutto. Dopo aver aggiornato Wine, **wineprefixcreate** può essere usato per aggiornare i dati del registro predefiniti.

Come abbiamo detto, potete modificare quei file .reg usando qualsiasi editor volete: assicuratevi soltanto che Wine non stia girando mentre lo fate, altrimenti tutti i cambiamenti andranno perduti.

Uso di regedit

Un modo più semplice per accedere e modificare il registro è con lo strumento `regedit`. Simile al programma di Windows che rimpiazza, regedit serve per dotare di una vista a livello di sistema sul registro che contiene tutte le chiavi. Avviate semplicemente *regedit* ed esso dovrebbe apparire. Noterete immediatamente che le misteriose chiavi mostrate nel file di testo sono organizzate in modo gerarchico.

Per navigare attraverso il registro cliccate i tasti sulla sinistra in modo da scendere ancora più in profondità. Per cancellare una chiave, cliccatela e scegliete "Delete" nel menu Edit. Per aggiungere una chiave o un valore, individuate dove volete metterli e scegliete "New" nel menu Edit. Allo stesso modo modificate una chiave esistente evidenziandola nella finestra a destra e scegliendo "Modify" nel menu Edit. Un'altra maniera per eseguire le stesse azioni è cliccare con il tasto destro sulla chiave o sul valore.

Particolarmente interessanti per gli utenti di Wine sono le impostazioni conservate in `HKEY_CURRENT_USER\Software\Wine`. Molte delle impostazioni che modificate con winecfg sono scritte in quest'area del registro.

Suggerimenti per l'amministrazione dei sistemi.

Con l'anzidetta struttura dei file, è possibile per un amministratore di sistema configurare il sistema in modo tale che una installazione Wine di sistema (e le sue applicazioni) possa esser condivisa da tutti gli utenti, pur lasciando a questi le loro configurazioni personalizzate. Un amministratore può, dopo aver installato Wine e qualsiasi software di applicazioni Windows che vuole accessibile agli utenti, copiare il relativo `system.reg` anche sopra i file di registro globali (che qui supponiamo trovarsi in `/usr/local/etc`), con:

```
cd ~root/.wine

cp system.reg /usr/local/etc/wine.systemreg
```

e forse anche collegare simbolicamente (*symlink*) questi all'account dell'amministratore, per semplificare successivamente l'installazione delle applicazioni nell'intero sistema:

```
ln -sf /usr/local/etc/wine.systemreg system.reg
```

Potreste essere tentati di fare altrettanto per `user.reg`, tuttavia tale file contiene delle impostazioni specifiche degli utenti. Ogni utente dovrebbe avere la propria copia di quel file insieme ai permessi per modificarlo.

Dovrete anche fare attenzione alle mappature dei drive. Se state condividendo il file system.reg, dovrete assicurarvi che le impostazioni del registro siano compatibili con la mappatura dei drive in `~/.wine/dosdevices` di ogni singolo utente. Come regola generale pratica, più mantenete la mappatura dei drive aderente alla configurazione predefinita fornita da **wineprefixcreate**, più facile sarà gestirla. Voi potreste essere in grado o meno di condividere alcuni o tutti gli effettivi drive "`c:`" su cui avete in origine installato le applicazioni. Alcune di esse richiedono la capacità di scrivere specifiche impostazioni nel drive, specialmente quelle progettate per Windows 95/98/ME.

Notate che lo script `tools/wineinstall,` utilizzato per qualcosa di ciò se avevate installato il sorgente di Wine in qualità di root, comunque ora non lo fa più.

Una parola finale per precauzione: fate attenzione a ciò che fate con l'account di amministratore - se copiate o collegate il registro dell'amministratore al registro globale, qualsiasi utente potrebbe leggere le preferenze dell'amministratore, il che potrebbe essere una pessima cosa se colà sono conservate delle informazioni sensibili (password, informazioni personali, ecc...). Usate l'account di amministratore solo per installare il software, non per il lavoro quotidiano: per quello usate un account di normale utente.

Elenco completo delle chiavi del registro

Troverete una lista aggiornata di utili chiavi e valori del registro nel wiki dello sviluppatore (http://wiki.winehq.org/UsefulRegistryKeys).

Altre cose da configurare

Questa sezione è stata intesa per trattare il resto delle cose che potete configurare. Serve anche come collezione di suggerimenti e trucchi per trarre il massimo dall'uso di Wine.

Porte seriali e parallele

La configurazione delle porte seriali e parallele è piuttosto simile a quella dei drive - create semplicemente un collegamento simbolico in `~/.wine/dosdevices` con il nome della periferica. Le porte seriali in Windows adottano una convenzione nella denominazione della parola "com" seguita da un numero, come `com1`, `com2`, ecc... Allo stesso modo le porte parallele usano "lpt" seguito da un numero, come `lpt1`. Dovreste collegare questi alle corrispondenti periferiche Unix, come /dev/ttyS0 e /dev/lp0. Ad esempio , per configurare una porta seriale ed una parallela, date i seguenti comandi:

```
ln -s /dev/ttyS0 com1
```

```
ln -s /dev/lp0 lpt1
```

Condivisioni di rete

Le condivisioni di Windows possono essere mappate nella directory `unc/` cosicché qualsiasi cosa provi ad accedere a `\\mioserver\qualcosa\file` guarderà in `~/.wine/dosdevices/unc/mioserver/qualcosa/file/`. Ad esempio, se usate Samba per montare `\\mioserver\qualcosa` su `/mnt/smb/mioserver/qualcosa` allora potete fare

```
ln -s /mnt/smb/mioserver/qualcosa /unc/mioserver/qualcosa
```

per renderlo disponibile in wine (non scordatevi di creare la directory unc se non esiste ancora).

Font

La configurazione dei font [tipi di caratteri], una volta un problema fastidioso, ora è molto più semplice. Se avete una collezione di file TrueType sotto Windows, basta semplicemente copiare i file `.ttf` in `c:\windows\fonts`.

Stampanti

Wine può interagire direttamente con il sistema di stampa CUPS locale per rendere disponibili le stampanti nel vostro sistema. Configurare le stampanti con Wine è semplice così come assicurarsi che la vostra configurazione CUPS funzioni. Wine necessita ancora del comando `lpr` (da CUPS) quando stampa un Documento.

Se non utilizzate CUPS, viene usato il vecchio sistema BSD-Printing:

·Tutte le stampanti in `/etc/printcap` vengono installate automaticamente in Wine.

·Wine ha bisogno di un file PPD per ciascuna stampante (generic.ppd viene dato con Wine)

·Il comando **`lpr`** viene chiamato al momento di stampare un documento

Scanner

Con Windows gli scanner usano la TWAIN API per accedere al sottostante hardware. L'integrata TWAIN DLL di Wine semplicemente redirige le richieste alle librerie SANE di Linux. Cosicché, per utilizzare i vostri scanner sotto Wine, dovrete prima assicurarvi di poter accedervi usando SANE. Dopo di che dovrete accertarvi di avere `xscanimage` pronto all'uso. Attualmente xscanimage è fornito con il pacchetto delle interfacce grafiche di sane, ma potrebbe non essere stato installato con la vostra distribuzione. Attualmente l'accesso agli scanner è noto per avere dei problemi. Se scoprite che vi funziona, per favore valutate di aggiornare questa sezione della guida dell'utente per fornire dettagli sull'uso di SANE con Wine.

Database ODBC

Il sistema ODBC di Wine, come il sistema di stampa, è progettato per connettersi al sistema unix ad alto livello. Piuttosto di assicurare che tutto il codice windows funzioni sotto wine, utilizza un provider Unix ODBC disponibile, come ad esempio UnixODBC. Così se configurate Wine per l'uso della odbc32.dll integrata, questa DLL di Wine si interfaccerà con il vostro pacchetto Unix

ODBC lasciando svolgere il lavoro a quello, mentre se configurate Wine per usare la odbc32.dll nativa, questa proverà ad utilizzare i gestori ODBC32, ecc...

Configurare ODBC in Unix

Il primo passo per utilizzare un sistema ODBC Unix con Wine è, naturalmente, di avere un sistema ODBC Unix funzionante. Ciò potrebbe implicare lo scaricamento di codice o RPM, ecc... Ci sono a disposizioni diversi sistemi ODBC per Unix: l'unico utilizzato dall'autore è unixODBC (con il gestore IBM DB2). Esiste anche and ODBC-ODBC bridge che può essere impiegato per accedere ad una base dati Microsoft Access. Di solito tali sistemi comprendono uno strumento, come **isql**, che vi consentirà di avere accesso ai dati tramite linea di comando in modo che possiate verificare che il sistema funzioni.

Il passo successivo è di agganciare la libreria ODBC Unix alla odbc32.DLL integrata in wine. quest'ultima (attualmente) cerca nella variabile ambientale *LIB_ODBC_DRIVER_MANAGER* il nome della libreria ODBC. Per esempio, nel file .bash.rc dell'autore c'è la linea:

```
export LIB_ODBC_DRIVER_MANAGER=/usr/lib/libodbc.so.1.0.0
```

Se quella variabile ambientale non è impostata, allora va in cerca di una libreria chiamata libodbc.so e così potete aggiungere un collegamento simbolico per equiparare quello alla vostra libreria. Per esempio, in qualità di root potreste lanciare i comandi:

```
$ ln -s libodbc.so.1.0.0 /usr/lib/libodbc.so
$ /sbin/ldconfig
```

L'ultimo passo nella configurazione di ciò è di assicurare che Wine sia impostato per avviare la versione integrata di odbc32.dll modificando la configurazione delle DLL. Questa DLL integrata si comporta semplicemente come tramite tra il codice chiamante e la libreria Unix ODBC.

Se incontrate qualche problema, allora potete usare il comando WINEDEBUG=+odbc prima di avvia wine per tracciare ciò che sta succedendo. Una parola di avvertimento. Alcuni programma in realtà imbrogliano un pochino e scavalcano la libreria ODBC. Per esempio il motore di Crystals Reports accede al registro per controllare il DSN. La correzione di questo è documentata nel sito di unixODBC in cui esiste una sezione sull'utilizzo di unixODBC con Wine.

Uso dei gestori ODBC di Windows

E' stato detto che i gestori nativi di ODBC funzionano con molti tipi di database, compresi MSSQL e Oracle. In effetti l'accesso da Linux ad alcuni come MSSQL può avvenire solo attraverso una applicazione di Winelib. Piuttosto di copiare le DLL soltanto, la maggioranza dei gestori ODBC richiede un installatore basato su Windows da avviare per configurare correttamente cose come le chiavi del registro.

Per impostare il supporto di MSSQL dovrete per prima cosa scaricare e far girare l'installatore mdac_typ.exe da microsoft.com. Poi, per configurare le Vostre connessioni ODBC, dovrete avviare CLICONFG.EXE e ODBCAD32.EXE sotto Wine. Potete trovare quest'ultimi nella directory windows\system dopo aver lanciato mdac_typ.exe. Confrontate i dati in uscita di questi programmi con quelli di una macchina nativa Windows. Alcune cose, come i protocolli, potrebbero mancare perché si suppone che vengano installate insieme al sistema operativo. Se è

così, potreste essere in grado di copiare le funzionalità assenti da un'installazione Windows esistente come pure qualsiasi valore di registro richiesto. Una installazione nativa Windows configurata per essere utilizzata da Wine dovrebbe funzionare nello stesso modo di quando funzionava nativamente.

Tipi provati con successo sotto Wine:

Tipo DB	**Utilizzabilità**
MS SQL	100,00%

Per favore segnalate qualsiasi altro accadimento alla lista postale di wine-devel (mailto:wine-devel@winehq.org).

Capitolo 4. Far funzionare Wine

Questo capitolo descriverà tutti gli aspetti del funzionamento di Wine, come ad es. la chiamata base di Wine, i parametri della linea di comando dei vari programmi di supporto di Wine, ecc...

Utilizzo di base: applicazioni e applet del pannello di controllo

Supponendo che stiate usando una installazione Windows fittizia, voi installate le applicazioni in Wine nella stessa maniera di Windows: avviando l'installatore [*installer*]. Potete accettare semplicemente i valori predefiniti di dove installare: molti installatori lo faranno in "C:\Programmi", che va bene. Se l'installatore dell'applicazione lo richiede, potreste scoprire che Wine crea icone nel vostro desktop e nel menu delle applicazioni. Se ciò succede, potete avviare le applicazioni cliccando su di esse.

Il modo normale per disinstallare qualcosa è per l'applicazione quello di mettere a disposizione un disinstallatore [*uninstaller*] normalmente registrato con l'applet dell pannello di controllo "Add/Remove Programs". Per accedere all'equivalente di Wine, avviate il programma **uninstaller** (è collocato nella directory `programs/uninstaller` in una directory del sorgente di Wine) in un *terminale*:

```
$ uninstaller
```

Alcuni programmi installano delle applet associate al pannello di controllo: esempi di ciò potrebbero essere Internet Explorer e QuickTime. Potete accedere al pannello di controllo avviando in un *terminale*:

```
$ wine control
```

che aprirà una finestra con dentro le applet del pannello di controllo installate, come in

Windows.

Se l'applicazione non installa nessuna voce nel menu o nel desktop, dovrete avviarla da linea di comando.

Ricordando dove l'avete installata, qualcosa come:

```
$ wine "c:\program files\nomeapplicazione\nomeapplicazione.exe"
```

probabilmente farà il trucchetto. Il percorso non è sensibile a maiuscole/minuscole, ma ricordatevi di mettere i doppi apici. Alcuni programmi non usano sempre una denominazione ovvia per le loro directory e i file .EXE, cosicché potreste dove cercare dentro la directory program files per vedere ciò che è stato messo dentro.

Come avviare Wine

Potete semplicemente invocare il comando **wine** per ottenere un breve messaggio di aiuto:

```
Wine 20040405
Usage: wine PROGRAM [ARGUMENTS...] Run the specified program
       wine --help                 Display this help and exit
```

```
wine --version              Output version information and exit
```

Il primo argomento dovrebbe essere il nome del file che volete che **wine** esegua. Se l'eseguibile si trova nella variabile ambientale `Path`, potete banalmente fornire il nome del file eseguibile. Tuttavia, se l'eseguibile non è in `Path`, allora dovete dare l'intero percorso dell'eseguibile (in formato Windows, non Unix!). Per esempio, posto il seguente `Path`:

```
Path="c:\windows;c:\windows\system;e:\;e:\test;f:\"
```

Potreste avviare il file `c:\windows\system\foo.exe` con:

```
$ wine foo.exe
```

Comunque dovreste avviare il file `c:\myapps\foo.exe` con questo comando:

```
$ wine c:\\myapps\\foo.exe
```

(notate la barra inversa "\" raddoppiata -*escaped*-!)

Per dettagli sull'avvio in modalità testo (CUI) degli eseguibili, leggete la sezione più avanti.

Ambienti grafici Wine simil-Explorer

Se preferite utilizzare un'interfaccia grafica per gestire i vostri file, potreste pensare di utilizzare Winefile. Questa applicazione di Winelib è fornita con Wine e può essere trovata con gli altri programmi di Wine. E' un modo utile per vedere la configurazione dei vostri drive, per localizzare i file ed in più potete eseguire i programmi direttamente da Winefile. Per favore osservate che molte funzioni non sono state ancora inserite.

Le opzioni a linea di comando di Wine

--help

Mostra una breve pagina di aiuto a linea di comando.

--version

Mostra la stringa con la versione di Wine. Utile per verificare la vostra installazione.

Variabili ambientali

WINEDEBUG=[canali]

Wine non è perfetto e molte applicazioni Windows ancora non funzionano senza errori sotto Wine (ma poi un mucchio di programmi non funzionano senza errori pure sotto il Windows nativo). Per semplificare alle persone la scoperta delle cause dietro a ciascun errore, Wine fornisce un certo numero di canali di correzione [*debug channels*] a cui potete ricorrere.

Ciascun canale di correzione, quando attivato, farà in modo che i messaggi di registro [*logging messages*] vengano mostrati nella console in cui avete invocato **wine**. Da qui potete redirigere i messaggi verso un file ed esaminare quest'ultimo con comodo. Ma state in allerta! Alcuni canali

di correzione possono generare volumi incredibili di messaggi di registro. Tra i "malfattori" più prolifici ci sono `relay`, che sputa fuori un messaggio di registro ogni volta che viene chiamata una funzione win32, `win`, che traccia lo scorrere dei messaggi di windows, e, naturalmente, `all`, che è un alias per ogni singolo canale di correzione esistente. Per un'applicazione complessa le vostre note di correzione [*debug log*] possono arrivare ad 1 MB o anche più. Un tracciamento di `relay` spesso genera oltre 10 MB di messaggi di registro, in base a quanto tempo fate funzionare l'applicazione (vorrete verificare la chiave di registro RelayExclude per variare ciò che riporta il tracciamento di `relay`). Le note rallentano un po' Wine, perciò non usate *WINEDEBUG* a meno che non ne abbiate realmente bisogno.

Con ogni canale di correzione potete inoltre specificare una *message class*, per filtrare le diverse gravità degli errori. Le quattro message class sono: `trace`, `fixme`, `warn`, `err`.

Per attivare un canale di correzione, usate la forma `class+channel`. Per disattivarlo, usate `class-channel`. Per elencare più di un canale nella stessa opzione *WINEDEBUG*, separatelincon virgole. Per esempio, per richiedere i messaggi della classe *warn* nel canale di correzione *heap*, potreste invocare **wine** in questo modo:

```
$ WINEDEBUG=warn+heap wine nome_programma
```

Se togliete la classe dei messaggi, **wine** mostrerà i messaggi di tutte le quattro classi di quel canale:

```
$ WINEDEBUG=heap wine nome_programma
```

Se volete vedere tutti i messaggi di registro [*log messages*] ad eccezione di quelli del canale relay, potreste fare qualcosa del genere:

```
$ WINEDEBUG=+all,-relay wine nome_programma
```

Ecco qui un elenco dei canali di correzione e delle classi di Wine. Altri canali saranno aggiunti (o tolti) nelle versioni successive.

Tabella 4-1. Canali di correzione

accel	adpcm	advapi	animate	aspi
atom	avicap	avifile	bidi	bitblt
bitmap	cabinet	capi	caret	cdrom
cfgmgr32	class	clipboard	clipping	combo
comboex	comm	commctrl	commdlg	computername
console	crtdll	crypt	curses	cursor
d3d	d3d_shader	d3d_surface	datetime	dc
ddeml	ddraw	ddraw_fps	ddraw_geom	ddraw_tex
debugstr	devenum	dialog	dinput	dll
dma	dmband	dmcompos	dmfile	dmfiledat
dmime	dmloader	dmscript	dmstyle	dmsynth
dmusic	dosfs	dosmem	dplay	dplayx
dpnhpast	driver	dsound	dsound3d	edit
enhmetafile	environ	event	eventlog	exec
file	fixup	font	fps	g711
gdi	global	glu	graphics	header

heap	hook	hotkey	icmp	icon
imagehlp	imagelist	imm	int	int21
int31	io	ipaddress	iphlpapi	jack
joystick	key	keyboard	listbox	listview
loaddll	local	mapi	mci	mcianim
mciavi	mcicda	mcimidi	mciwave	mdi
menu	menubuilder	message	metafile	midi
mmaux	mmio	mmsys	mmtime	module
monthcal	mpeg3	mpr	msacm	msdmo
msg	mshtml	msi	msimg32	msisys
msrle32	msvcrt	msvideo	mswsock	nativefont
netapi32	netbios	nls	nonclient	ntdll
odbc	ole	oledlg	olerelay	opengl
pager	palette	pidl	powermgnt	print
process	profile	progress	propsheet	psapi
psdrv	qcap	quartz	ras	rebar
reg	region	relay	resource	richedit
rundll32	sblaster	scroll	seh	selector
server	setupapi	shdocvw	shell	shlctrl
snmpapi	snoop	sound	static	statusbar
storage	stress	string	syscolor	system
tab	tape	tapi	task	text
thread	thunk	tid	timer	toolbar
toolhelp	tooltips	trackbar	treeview	ttydrv
twain	typelib	uninstaller	updown	urlmon
uxtheme	ver	virtual	vxd	wave
wc_font	win	win32	wineboot	winecfg
wineconsole	wine_d3d	winevdm	wing	winhelp
wininet	winmm	winsock	winspool	wintab
wintab32	wnet	x11drv	x11settings	xdnd
xrandr	xrender	xvidmode		

Per maggiori dettagli sui canali di correzione, controllate "The Wine Developer's Guide" (http://www.winehq.org/site/docs/winedev-guide).

WINEDLLOVERRIDES=[DLL Overrides]

Non è sempre possibile avviare un'applicazione con le DLL integrate. Alle volte le DLL native funzionano semplicemente meglio. Sebbene queste eccezioni [*overrides*] delle DLL possano essere impostati con l'utilizzo di winecfg, potreste voler usare la variabile WINEDLLOVERRIDES per definirli.

Ad esempio, se voleste che wine utilizzi le native ole32.dll, oleaut32.dll e rpcrt4, potreste avviare **wine** in questo modo:

```
$ WINEDLLOVERRIDES="ole32.dll, oleaut32.dll, rpcrt4=n" wine nome_programma
```

Per maggiori informazioni sulle eccezioni delle DLL, fate riferimento per favore alla sezione delle eccezioni delle DLL di questa guida.

Impostazioni del gestore audio OSS

Se state usando il gestore audio [*audio driver*] OSS ed avete molte periferiche (ad es. /dev/dsp*, /dev/mixer*), con le seguenti variabili ambientali potete specificare quella che volete utilizzare:

- AUDIODEV=[periferica audio]
- MIXERDEV=[periferica mixer]
- MIDIDEV=[periferica MIDI]

Quale esempio:

```
$ AUDIODEV=/dev/dsp4 MIXERDEV=/dev/mixer1 MIDIDEV=/dev/midi3 wine nome_programma
```

Opzioni a linea di comando di wineserver

Di solito wineserver viene avviato automaticamente da Wine ogni volta che viene fatto partire il primo processo di wine. Tuttavia wineserver ha alcune utili opzioni a linea di comando che potete aggiungere se lo avviate manualmente, ad es. tramite uno script di login dell'utente o similmente.

`-d<n>`

Imposta il livello di correzione per l'emissione dei dati di correzione nel terminale in cui è stato avviato wineserver a livello <n>. In altri termini: ogni cosa più grande di 0 abiliterà l'emissione degl specifici dati di correzione di wineserver.

`-h`

Mostra un messaggio di aiuto sulle opzioni a linea di comando di wineserver.

`-k[n]`

Uccide il wineserver corrente, eventualmente con il segnale `n`.

`-p[n]`

Questo segnale rende persistente wineserver, eventualmente per n secondi. Impedisce la chiusura immediata di wineserver.

Di solito wineserver cessa quasi immediatamente dopo che è stato chiuso l'ultimo processo che ne fa uso. Tuttavia, dal momento che wineserver carica un mucchio di cose all'avvio (come tutti i dati dell'intero registro di Windows), la sua partenza potrebbe essere così lenta che risulta molto utile mantenerlo attivo dopo il termine di tutte le sessioni di Wine rendendolo persistente.

`-w`

Questo parametro fa attendere a wineserver nuovamente riavviato fintanto che la corrente istanza attiva di wineserver cessa.

Definizione delle variabili ambientali di Windows/DOS

Il vostro programma potrebbe richiedere la precisa definizione di alcune variabili ambientali per funzionare con successo. In tal caso dovete impostare queste variabili nella shell di Linux, dal momento che Wine passerà l'intera definizione delle variabili ambientali allo spazio Windows dedicato ad esse. Un esempio per la shell bash (altre shell potrebbero avere una sintassi differente!);

```
export MYENVIRONMENTVAR=myenvironmentvarsetting
```

Ciò assicurerà che il vostro programma Windows possa accedere alla variabile ambientale MYENVIRONMENTVAR una volta che l'abbiate avviato con Wine. Se desiderate avere MYENVIRONMENTVAR definita in modo permanente, allora potete collocare la definizione in `/etc/profile` od anche in `~/.bashrc` nel caso di bash.

Tenete presente tuttavia che esistono delle eccezioni alla regola: se volete cambiare le variabili PATH, SYSTEM o TEMP, ovviamente non potete modificarle in questo modo poiché ciò altererebbe le impostazioni dell'ambiente Unix. Invece dovreste impostarle nel registro: per farlo dovreste lanciare **`wine regedit`** e poi andare nella chiave

```
HKEY_CURRENT_USER/Environment
```

Ora potrete creare o modificare i valori delle variabili che vi servono.

```
"System" = "c:\\windows\\system"
```

Questa definisce dove si trovano i file di sistema. La directory di sistema Windows dovrebbe risiedere sotto la directory utilizzata per le impostazioni di `Windows`. Così quando si usa `/usr/local/wine_c_windows` come percorso di Windows, la directory dovrebbe essere `/usr/local/wine_c/windows/system`. Deve essere definita senza la barra normale e dovete sincerarvi di avere l'accesso in scrittura ad essa.

```
"Temp" = "c:\\temp"
```

Questa dovrebbe essere la directory dove volote che siano riposti i vostei file temporanei, `/usr/local/wine_c/temp` nel nostro esempio precedente. Di nuovo: niente sbarra normale ed *accesso in scrittura*!!

```
"Path" = "c:\\windows;c:\\windows\\system;c:\\blanco"
```

si comporta come la definizione di PATH nelle macchine UNIX. Quando wine viene avviato come nel caso **`wine sol.exe`**, se `sol.exe` risiede in una directory specificata nella definizione di `Path`, wine lo farà girare (naturalmente anche se `sol.exe` si trova nella directory corrente, wine lo farà funzionare). Accertatevi che abbia le vostre directory windows e di sistema (per questa impostazione deve avere `"c:\\windows;c:\\windows\\system"`).

Programmi in modalità testo (CUI: Console User Interface)

I programmi in modalità testuale sono programmi i cui risultati sono rappresentati da testo (sorpresa!). Nella terminologia di Windows sono chiamati eseguibili CUI (Console User Interface) invece che eseguibili GUI (Graphical User Interface). Win32 API fornisce un insieme

completo di APi per gestire questa situazione, che va dalle funzionalità basilari, come la stampa di testi, fino a quelle di alto livello (come la scrittura a schermo intero, il supporto dei colori, il movimento del cursore, il supporto del mouse), passando attraverso funzionalità come la scrittura su linea o il supporto di flussi in ingresso grezzi/elaborati.

Data l'estesa varietà delle precedenti funzionalità e l'attuale utilizzo nel mondo Un*x, Wine viene fornito con tre diverse modalità per l'avvio di un programma da console (altrimenti detto eseguibile CUI):

- flussi puri (bare streams)
- wineconsole con backend dell'utente
- wineconsole con curses backend

I nomi qui sono un pochino oscuri. "bare streams" significa che non viene fornito alcun supporto extra di wine per la mappatura a l'accesso alla console di unix e quella Windows. Le altre due modalità richiedono l'uso di uno specifico programma di Wine (wineconsole) che fornisce strutture estese. La tabella seguente descrive che cosa potete fare (e non fare) con quelle tre modalità.

Tabella 4-2. Differenze di base tra console

Funzione	**Semplici flussi**	**Wineconsole & backend utente**	**Wineconsole & curses backend**
Come avviare (supponendo che l'eseguibile si chiami foo.exe)	**`$ wine foo.exe`**	**`$ wineconsole -- --backend=user foo.exe`**	**`$ wineconsole foo.exe nd=user foo.exe`** Potete anche usare **`-- backend=curses`** come opzione
Buon supporto alle applicazioni CUI orientate alle linea	Si	Si	Si
Buon supporto per le applicazioni CUI a schermo intero (supporto compreso ma illimitato al colore, supporto al mouse, ...)	No	Si	Si
Può funzionare anche se X11 non sta girando	Si	No	Si
Implementazione	Mappa i flussi Windows standard in quelli standard Unix (stdin/stdout/stderr)	Wineconsole creerà una nuova finestra (quindi richiedendo la disponibilità della	Wineconsole userà la console unix esistente (da cui è stato avviato il programma) e con

Funzione	**Semplici flussi**	**Wineconsole & backend utente**	**Wineconsole & curses backend**
		USER32 DLL) dove saranno mostrate tutte le informazioni	l'aiuto della libreria (n)curses prenderà il controllo di tutta la superficie del terminale per interagire con l'utente
Limiti noti			Produrrà uno strano comportamento se due (o più) console Windows vengono utilizzate nello stesso terminale Un*x.

Configurazione degli eseguibili CUI

Quando si utilizza wineconsole sono disponibili diverse opzioni di configurazione. Wine (come fa Windows) conserva, in base alla singola applicazione, diverse opzioni nel registro. Ciò lascia, per esempio, ad un utente la definizione della dimensione del buffer dello schermo che preferisce avere per una data applicazione.

Attualmente soltanto il backend USER vi permette di editare quelle opzioni (raccomandiamo di non editare a mano i contenuti del registro). Questa edizione viene avviata immediatamente quando un utente clicca col tasto di destra nella console (ciò fa apparire un menu) in cui potete scegliere tra:

- Default: ciò editerà le impostazioni condivise da tutte le applicazioni che non sono state ancora configurate. Così, quando per la prima volta (nella vostra macchina, sotto il vostro account) viene avviata in wineconsole, quest'ultimo erediterà tali impostazioni predefinite per l'applicazione. Dopo l'applicazione avrà le proprie impostazioni che sarete in grado di modificare a piacimento.
- Properties: questo modificherà le impostazioni dell'applicazione. Quando l'avrete fatto, con le modifiche, vi sarà richiesto se volete:
    1. Mantenere le impostazioni modificate solo per questa sessione (la prossima volta che avvierete l'applicazione, non troverete le modifiche appena effettuate).
    2. Usare le impostazioni ed anche salvarle, cosicché la prossima volta che avvierete l'applicazione, troverete ancora le nuove impostazioni.

Qui c'è un elenco delle voci che potete configurare con il loro significato:

Tabella 4-3. Opzioni della configurazione di wineconsole

Opzioni di configurazione	**Significato**
Cursor's size	Definisce la dimensione del cursore. Sono

Opzioni di configurazione	**Significato**
	disponibili tre opzioni: small (33% dell'altezza del carattere), medium (66% dell'altezza del carattere) e large (100%)
Popup menu	Prima si è detto che la comparsa della configurazione di wineconsole si attiva cliccando il tasto destro nella finestra della console. Tuttavia potrebbe essere un problema quando l'applicazione che avete avviato in wineconsole attende gli eventi del clic destro per inviarglieli. Marcando control o shift selezionate dei modificatori aggiuntivi al clic destro per l'apertura del menu a scomparsa. Ad esempio, spuntando shift, gli eventi verranno inviati all'applicazione quando cliccherete col tasto destro senza premere il tasto shift [maiuscolo] ed [invece] aprirete la finestra cliccando col tasto destro mentre premete shift.
Quick edit	Questa casella di spunta vi lascia decidere se gli eventi del tasto sinistro del mouse dovranno essere interpretati come eventi da inviare all'applicazione sottostante (tick off o disattivato) oppure come selezione di un'area rettangolare dello schermo da copia successivamente nella clipboard (tick on o attivato).
History	Ciò vi lascia scegliere quanti comandi desiderate richiamare da console. Potete anche sia controllare a vostro piacimento quando inserire diverse volte lo stesso comando - potenzialmente inframezzato con altri - sia se preferite conservarli tutti (niente spunta) o soltanto l'ultimo (spunta).
Police	La scheda della proprietà Police permette di selezionare il carattere predefinito per la console (file dei font, misura, sfondo e colore in primo piano).
Screenbuffer & window size	La console così come la vedete è costituita di due parti differenti. Da un lato c'è lo screenbuffer [buffer dello schermo] che

Opzioni di configurazione	**Significato**
	contiene tutte le informazioni che la vostra applicazione manda allo schermo, e dall'altro la finestra (*window*) che mostra una determinata area di questo *screenbuffer*. Notate che la finestra è sempre più piccola o della stessa dimensione del buffer di schermo. Tenere rigorosamente una dimensione più piccola metterà le sbarre di scorrimento nella finestra in modo che possiate vedere l'intero contenuto del buffer di schermo.
Close on exit	Se è selezionato, allora wineconsole terminerà alla chiusura dell'applicazione. Altrimenti rimarrà aperto finché l'utente non lo chiuda manualmente: ciò pe mette di vedere le ultime informazioni di un programma dopo che è stato terminato.
Edition mode	Quando l'utente inserisce dei comandi, può scegliere tra alcuni modi di edizione (*edition modes*): • Emacs: sono disponibili le stesse combinazioni di tasti di emacs. Per esempio Ctrl-A condurrà il cursore all'inizio della linea da modificare. Guardate il vostro manuale di emacs per i dettagli sui comandi. • Win32: queste sono le combinazioni standard della console di Windows (che usa principalmente le frecce).

Capitolo 5. Eliminazione dei problemi / segnalazione degli errori

Cosa fare se un qualche programma non funziona ancora?

Ci sono volte che avete provato di tutto, avete pure ucciso un gatto durante la luna piena e l'avete divorato con aglio marcio e pesce guasto ballando la Danza del Demonio, eppure ancora nulla vi è stato di aiuto per far funzionare qualche dannato programma in una qualche versione di Wine. Non disperate, siamo qui per aiutarvi... (in altri termini: quanto siete disposti a pagare?).

Verificate la vostra configurazione di wine

Osservate il risultato di `$ wine --version` per assicurarvi che state eseguendo una versione recente di wine. Lanciate winecfg e scorrete le impostazioni per accertarvi che sembrino normali. Guardate in `~/.wine/dosdevices` per essere sicuri che `c:` punti dove ritenete che dovrebbe.

Utilizzate diverse impostazioni della versione di windows

In alcuni casi l'uso di impostazioni differenti di windows può giovare.

Usate percorsi d'avvio differenti

Pure questo talvolta aiuta. Provate ad usare sia **wine prg.exe**, sia **wine x:\\full\\path\\to\\prg.exe**

Giocate con la configurazione delle DLL

Avviate con `WINEDEBUG=+loaddll` per capire quali DLL vengono usate e se sono state caricate come native o integrate. Poi accertatevi di avere i giusti file delle DLL native nella vostra configurata directory C:\windows\system e giocate con le impostazioni dell'ordine di caricamento delle DLL a linea di comando o nel file di configurazione.

Controllate il vostro ambiente di sistema!

Solo un'idea: potrebbe essere che il vostro ambiente Wine di costruzione/esecuzione sia corrotto? Accertatevi che non ci siano problemi di qualsiasi genere con i pacchetti da cui dipende Wine (gcc, glibc, librerie X, OpenGL(!), ...). Ad es. alcune persone vanno incontro a strani guasti nel trovare roba quando usano file di intestazione "sbagliati" per le librerie "giuste" (cosa che deriva da giorni di caccia all'errore nel disperato tentativo di scoprire perché quella funzione a basso livello fallisca in un modo che totalmente al di là dell'immaginabile... ARGH!).

Usate modalità di GUI (Window Manager) differenti

Istruite Wine tramite file di configurazione ad utilizzare entrambe le modalità di desktop, sia quella gestita,sia quella "normale" semplice e brutta. Anche questo può rendere un inferno unico

per una semplice differenza.

Controllate la vostra applicazione!

Forse la vostra applicazione sta utilizzando un qualche tipo di protezione dalla copia? Attualmente molte protezioni dalla copia non funzionano con Wine: alcune potrebbero però funzionare in futuro (il livello CD-ROM non è in realtà ancora pienamente realizzato).

Andate su GameCopyWorld (http://www.gamecopyworld.com) e provate a trovare una sprotezione decente per il vostro gioco che si sbarazzi di quella brutta protezione. Spero che voi abbiate una copia legale del programma, tuttavia... ;-)

Controllate il vostro ambiente Wine!

Funzionare con o senza una partizione Windows può avere un impatto dirompente. Configurate Wine per fare l'opposto di quello che siete soliti avere. Installate anche DCOM98 o DCOM95: ciò potrebbe essere di grande beneficio.

Riconfigurate Wine

Talvolta il processo di installazione di wine cambia e le nuove versioni di Wine fanno riferimento a questi cambiamenti. Questo è vero in special modo se la configurazione iniziale è stata realizzata molto tempo fa. Rinominate la vostra directory esistente `~/.wine` a scopo di sicurezza. Utilizzate il processo di installazione raccomandato dalla vostra distribuzione per creare una nuova configurazione. Usate le informazioni della vecchia directory `~/.wine` come riferimento. Per quanto riguarda la distribuzione di wine sorgente, per configurare Wine avviate lo script `tool/wineinstall` come utente se volete andare a buon fine. Questa è un'operazione abbastanza sicura. Più tardi potete rimuovere la nuova directory `~/.wine` e rinominare nuovamente quella vecchia.

Controllate ulteriori informazione

Esiste veramente una buona possibilità che qualcuno abbia già provato a fare la stessa cosa vostra. Potreste trovare utili le seguenti risorse:

- Esplorate il WineHQ's Application Database (http://appdb.winehq.org) per verificare ogni suggerimento relativo al programma. Se la vostra specifica versione del programma non è elencata, potreste trovarne una differente che contiene abbastanza informazioni da aiutarvi a venirne fuori.
- Frank's Corner (http://www.frankscorner.org) contiene un elenco di applicazioni e di informazioni dettagliate per impostarle. Ulteriore aiuto si può trovare nei forum degli utenti.
- Google (http://www.google.com) può risultare utile in base a come lo utilizzate. Potreste trovarlo utile per cercare in Google Groups (http://groups.google.com), in particolare nel gruppo comp.emulators.ms-windows.wine (http;//groups.google.com/groups?hl=en&lr=&ie=UTF-8&group= comp.emulators.ms-windows.wine).
- Freenode.net (http://www.freenode.net) ospita un canale IRC per Wine. Potete accedervi utilizzando qualsiasi cliente IRC come ad esempio Xchat. Le impostazioni di cui avete bisogno sono: server=irc.freenode.net, porta=6667 e canale=#winehq

- Se avete un programma che abbisogna dell'ambiente runtime di Visual Basic [Visual Basic Runtime Environment], tete scaricarlo dal sito Microsoft (http://www.microsoft.com/downloads/details.aspx?FamilyID=bf9a24f9-b5c5-48f4-8edd-cdf2d29a79d5&DisplayLang=en/)
- Se sapete che vi manca una DLL, come ad esempio mfc42, potete essere in grado di rintracciarla su www.dll-files.com (http://www.dll-files.com/)
- Anche le liste postali di Wine (http://www.winehq.org/site/forums#ml) possono essere d'aiuto, in particolare wine-users. La lista wine-devel può essere appropriata in base al tipo di problema che state incontrando. Se inviate messaggi a wine-devel, dovreste essere pronti ad un piccolo sforzo per aiutare a diagnosticare il problema. Leggete la sezione di sotto per scoprire come eliminare gli errori dalla fonte del vostro problema.
- Se tutto il resto fallisce, potreste decidere di rivolgervi alle versioni commerciali di Wine per vedere se la vostra applicazione è supportata.

Correggetelo!

Scoprire la fonte del vostro problema è il passo successivo da intraprendere. Esiste un ampio spettro di problemi che vanno dalle semplici questioni di configurazione alle funzionalità totalmente non ancora inserite in Wine. La prossima sezione descriverà come raccogliere un rapporto di errore e come correggere un blocco. Per maggiori informazioni sull'impiego dei mezzi di correzione, assicuratevi di leggere la Wine Developers Guide [Guida per gli sviluppatori di Wine].

Come segnalare un errore

Per favore segnalate tutti gli errori assieme a qualsiasi informazione rilevante a Wine Bugzilla (http://bugs.winehq.org/). Per favore consultate la base dati di Bugzilla per verificare se il vostro problema è già segnalato. Se così fosse, per favore aggiungete qualsiasi informazione rilevante all'originale rapporto d'errore.

Tutti i rapporti di errore

Alcuni semplici consigli per rendere più utile il vostro rapporto di errore

(e per avere maggiori probabilità di risposte e correzioni):

1. Inviate informazioni rilevanti più che potete.

   Ciò significa che abbiamo bisogno di maggiori informazioni rispetto ad una semplice "MS Word si blocca ogni volta che lo avvio. Sapete perché?". Almeno includete le seguenti informazioni:

   - Quale versione di Wine state utilizzando (avviare **wine --version**)
   - Il nome del Sistema Operativo che state usando, quale distribuzione (se del caso) e quale versione (ad es. Linux Red Hat 7.2)
   - Quale compilatore e versione (avviate **gcc -v**). Se non compilate Wine, allora il

nome del pacchetto e dove l'avete preso.

- La versione di Windows, se utilizzata con Wine. Menzionate se non usate Windows.
- Il nome del programma che state tentando di avviare, il suo numero di versione ed un URL per dove ci si può procurare il programma (se disponibile).
- L'esatta linea di comando che avete usato per avviare wine (ad es. **wine "C:\Programmi\Test\programma.exe"**).
- I passi esatti richiesti per riprodurre l'errore.
- Qualsiasi altra informazione che pensate possa essere rilevante o utile, come la versione del server X in caso di problemi con X, versione di libc, ecc...

2. Riavviate il programma con l'opzione `WINEDEBUG=+relay` della variabile ambientale WINEDEBUG (cioè **WINEDEBUG=+relay wine sol.exe**).

Ciò genererà nella console delle informazioni aggiuntive che potrebbero essere utili nella correzione del programma. Tale opzione rallenta pure l'esecuzione del programma. Esistono alcuni casi in cui l'errore sembra sparire quando si utilizza `+relay`: per favore segnalatelo nel rapporto d'errore.

Arresti

Se Wine si arresta mentre avvia il vostro programma, è importante che noi abbiamo questa informazione per consentirci di comprendere ciò che sta causando il blocco (*crash*). Questo può genera parecchie informazioni (diversi MB), sicché è meglio produrlo su file. Quando appare l'invito (*prompt*) `Wine-dbg>`, battete **`quit`**.

Potreste voler provare `+relay`, `+snoop` invece di `+relay`, ma per favore notate che `+snoop` è piuttosto instabile e spesso andrà in blocco prima di un semplice `+relay`! Se questo è il caso, per favore usate esclusivamente `+relay`!! Un rapporto d'errore con un blocco da codice `+snoop` in molti casi è inutile! Potete attivare altri parametri in base al problema su cui state investigando. Guardate la pagina man di wine per un elenco completo dei parametri.

Per ottenere i risultati di trace, utilizzate uno dei metodi seguenti:

Il modo semplice

1. Questo metodo è destinato a permettere anche ad un totale principiante di inviare un rilevante registro di trace (*trace log*) in caso di blocco.

   Per far funzionare questo metodo il vostro computer *deve* avere installato perl. Per scoprire se avete perl, lanciate **which perl**: se questo restituisce qualcosa come `/user/bin/perl`, siete a cavallo. Altrimenti passate oltre a "Il modo complicato". Se non siete sicuri, continuate ad andare avanti. Quando proverete ad avviare lo script, esso diverrà *molto* evidente se non avrete perl.
2. Cambiate la directory in `<dir_di_wine>/tools`
3. Inserite **./bug_report.pl** e seguite le istruzioni.
4. Inviate l'errore a Wine Bugzilla (http://bugs.winehq.org/). Per favore, prima di impostare un rapporto d'errore, esaminate la base dati Bugzilla per controllare se il vostro problema è già

stato riscontrato. Includete la vostra dettagliata descrizione personale del problema con informazioni rilevanti. Allegate il "Nice Formatted Report" all'errore inviato. Non tagliate e cucite il rapporto nella descrizione dell'errore - è piuttosto grande. Conservate l'intero risultato della correzione nel caso serva agli sviluppatori di Wine.

Il modo complicato

E' probabile che solo le ultime 100 o quasi linee del trace siano necessarie per scoprire dove si blocca il programma. Per ottenere queste ultime 100 linee abbiamo bisogno di fare ciò che segue:

1. Redirigere i risultati di *WINEDEBUG* su un file
2. Separare le ultime 100 linee verso un altro file utilizzando **tail**.

Questo si può fare utilizzando uno dei metodi seguenti.

Per tutte le shell:

```
     $ echo quit | WINEDEBUG=+relay wine [altre_opzioni]
nome_programma >& nomefile.out;
```

```
$ tail -n 100 nomefile.out > file_rapporto
```

(ciò stamperà i rapporti di correzione di wine solo nel file e poi si fermerà da solo. Probabilmente è una buona idea utilizzare questo comando dal momento che wine stampa così tanti messaggi di correzione che quest'ultimi intaserebbero il terminale, divorando cicli di CPU).

Per tcsh e le altre shell simili a csh:

```
     $ WINEDEBUG=+relay wine [altre_opzioni] nome_programma |&
tee nomefile.out;
```

```
$ tail -n 100 nomefile.out > file_rapporto
```

Per bash e le altre shell simili a sh:

```
     $ WINEDEBUG=+relay wine [altre_opzioni] nome_programma 2>&1
tee nomefile.out;
```

```
$ tail -n 100 nomefile.out > file_rapporto
```

Ora `file_rapporto` conterrà le ultime cento linee del risultato della correzione, comprese la copia del registro e il backtrace [retrotracciamento], che sono le porzioni più importanti delle informazioni. Per favore non cancellate questa parte, anche se non capite cosa significhi.

Inviate l'errore a Wine Bugzilla (http://bugs.winehq.org). Dovete allegare il file risultante `file_rapporto` dalla parte 2) assieme alle informazioni rilevanti usate per crearlo. Non tagliate e cucite il rapporto nella descrizione dell'errore - è piuttosto grande e creerebbe un pasticcio nel rapporto d'errore. Se fate ciò, le vostre possibilità di ricevere un qualche tipo di utile risposta dovrebbero essere piuttosto buone.

Per favore, esaminate la base dati Bugzilla per verificare se il vostro problema è già stato segnalato. Se così fosse, allegate il file risultante `file_rapporto` all'originale rapporto d'errore e aggiungete qualsiasi altra informazione rilevante.

Glossario

Binario

Un file che è in forma compilata, eseguibile nella macchina: dati esadecimali [*hex*] (al contrario di un file sorgente).

Codice sorgente

Il codice sorgente (*source code*) è il codice di cui consiste un programma prima di essere compilato, cioè le istruzioni originali di creazione di un programma che dicono ad un compilatore come dovrebbe apparire un programma una volta compilato in un *Binario*.

Distribuzione

Di solito una distribuzione è il modo in cui alcuni "venditori" forniscono i CD di sistemi operativi (abitualmente menzionata nel contesto di Linux). Un ambiente Linux può essere fornito in molte configurazioni differenti: ad es. delle distribuzioni potrebbero essere realizzate per essere adatte ai giochi, ad applicazioni scientifiche, ad operazioni di server, a sistemi desktop, ecc...

DLL

Una DLL (Dynamic Link Library o libreria a collegamento dinamico) è un file che può essere caricato ed eseguito dinamicamente da dei programmi. Sostanzialmente si tratta di un deposito di codice esterno per programmi. Dal momento che di solito diversi programmi riutilizzano la stessa DLL invece di avere quel codice nei propri file, ciò riduce drasticamente lo spazio di achiviazione richiesto. Un sinonimo di DLL dovrebbe essere libreria.

Editor

Solitamente un editor è un programma per creare o modificare file di testo. Sotto Linux sono disponibili svariati editor in modalità grafica e testo.

Esempi di editor grafici sono: nedit, gedit, kedit, xemacs, gxedit.

Esempi di editor testuali sono: joe, ae, emacs, vim, vi. In un *terminale* avviateli semplicemente tramite:

```
$ nomeeditor nomefile
```

Git

Git è un veloce gestore di contenuti di directory (*directory content manager*), scritto in origine per l'impiego in grandi depositi (*repository*), come il sorgente del kernel di Linux. Guardate il capitolo di Git nella "Wine Developer Guide" per dettagliate informazioni sull'uso.

Pacchetto

Un pacchetto è un file compresso in uno specifico formato della *distribuzione*. Contiene i file di uno specifico programma che volete installare. I pacchetti vengono solitamente installati tramite i gestori di pacchetti **dpkg** o **rpm**.

root

root (radice) è il nome dell'account dell'amministratore di sistema. Per avviare i programmi come

root, aprite semplicemente una finestra di *Terminale* e poi avviate:

```
$ su -
```

Questo vi domanderà la password dell'utente root del vostro sistema e, fatto ciò, sarete in grado di svolgere i compiti di amministrazione del sistema che richiedono gli speciali privilegi di root. L'account di root viene indicato con l'invito (*prompt*)

```
#
```

mentre '$' indica l'account di un utente normale.

Shell

Una shell è uno strumento che permette agli utenti di interagire con il sistema. Solitamente le shell sono basate su testo ed orientate alla linea di comando. Esempi di shell diffuse includono **bash**, **tcsh** e **ksh**. Wine suppone che per le operazioni di installazione di Wine voi usiate **bash**, dal momento che questa è la shell più diffusa sotto Linux. Le shell abitualmente girano in una finestra di *Terminale*.

Terminale

Una finestra di terminale è abitualmente una finestra grafica che uno utilizza per eseguire una ***Shell***. Se Wine vi chiede di aprire un terminale, allora voi di solito dovete cliccare su una icona del vostro desktop che mostra una grande finestra nera (o, in altri casi, una icona che presenta una conchiglia marina [*shell* in inglese significa conchiglia - ndt.]). Wine suppone che stiate usando la shell **bash** in una finestra di terminale, cosicché, se capita che il vostro terminale utilizzi un programma di shell differente, battete semplicemente:

```
bash
```

nella finestra di terminale.

Variabile ambientale

Le variabili ambientali sono definizioni testuali utilizzate da una *Shell* per conservare delle importanti impostazioni di sistema. In una shell **bash** (quella utilizzata più comunemente in Linux) potete vedere tutte le variabili eseguendo:

```
set
```

Se volete cambiare una variabile ambientale, potete avviare:

```
export MIAVARIABILE=miocontenuto
```

Per cancellare una variabile ambientale usate:

```
unset MIAVARIABILE
```

Wine License

Copyright (c) 1993-2005 the Wine project authors (see the file AUTHORS for a complete list)

Wine is free software; you can redistribute it and/or modify it under the terms of the GNU Lesser General Public License as published by the Free Software Foundation; either version 2.1 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU Lesser General Public License for more details.

A copy of the GNU Lesser General Public License is included in the Wine distribution in the file COPYING.LIB. If you did not receive this copy, write to the Free Software Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA.

GNU Lesser General Public License

Version 2.1, February 1999

```
Copyright (C) 1991, 1999 Free Software Foundation, Inc.
59 Temple Place, Suite 330, Boston, MA  02111-1307  USA
Everyone is permitted to copy and distribute verbatim copies
of this license document, but changing it is not allowed.

[This is the first released version of the Lesser GPL.  It also counts
 as the successor of the GNU Library Public License, version 2, hence
 the version number 2.1.]
```

Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public Licenses are intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users.

This license, the Lesser General Public License, applies to some specially designated software packages--typically libraries--of the Free Software Foundation and other authors who decide to use it. You can use it too, but we suggest you first think carefully about whether this license or the ordinary General Public License is the better strategy to use in any particular case, based on the explanations below.

When we speak of free software, we are referring to freedom of use, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish); that you receive source code or can get it if you want it; that you can change the software and use pieces of it in new free programs; and that you are informed that you can do these things.

To protect your rights, we need to make restrictions that forbid distributors to deny you these rights or to ask you to surrender these rights. These restrictions translate to certain responsibilities for you if you distribute copies of the library or if you modify it.

For example, if you distribute copies of the library, whether gratis or for a fee, you must give the

recipients all the rights that we gave you. You must make sure that they, too, receive or can get the source code. If you link other code with the library, you must provide complete object files to the recipients, so that they can relink them with the library after making changes to the library and recompiling it. And you must show them these terms so they know their rights.

We protect your rights with a two-step method: (1) we copyright the library, and (2) we offer you this license, which gives you legal permission to copy, distribute and/or modify the library.

To protect each distributor, we want to make it very clear that there is no warranty for the free library. Also, if the library is modified by someone else and passed on, the recipients should know that what they have is not the original version, so that the original author's reputation will not be affected by problems that might be introduced by others.

Finally, software patents pose a constant threat to the existence of any free program. We wish to make sure that a company cannot effectively restrict the users of a free program by obtaining a restrictive license from a patent holder. Therefore, we insist that any patent license obtained for a version of the library must be consistent with the full freedom of use specified in this license.

Most GNU software, including some libraries, is covered by the ordinary GNU General Public License. This license, the GNU Lesser General Public License, applies to certain designated libraries, and is quite different from the ordinary General Public License. We use this license for certain libraries in order to permit linking those libraries into non-free programs.

When a program is linked with a library, whether statically or using a shared library, the combination of the two is legally speaking a combined work, a derivative of the original library. The ordinary General Public License therefore permits such linking only if the entire combination fits its criteria of freedom. The Lesser General Public License permits more lax criteria for linking other code with the library.

We call this license the "Lesser" General Public License because it does Less to protect the user's freedom than the ordinary General Public License. It also provides other free software developers Less of an advantage over competing non-free programs. These disadvantages are the reason we use the ordinary General Public License for many libraries. However, the Lesser license provides advantages in certain special circumstances.

For example, on rare occasions, there may be a special need to encourage the widest possible use of a certain library, so that it becomes a de-facto standard. To achieve this, non-free programs must be allowed to use the library. A more frequent case is that a free library does the same job as widely used non-free libraries. In this case, there is little to gain by limiting the free library to free software only, so we use the Lesser General Public License.

In other cases, permission to use a particular library in non-free programs enables a greater number of people to use a large body of free software. For example, permission to use the GNU C Library in non-free programs enables many more people to use the whole GNU operating system, as well as its variant, the GNU/Linux operating system.

Although the Lesser General Public License is Less protective of the users' freedom, it does ensure that the user of a program that is linked with the Library has the freedom and the wherewithal to run that program using a modified version of the Library.

The precise terms and conditions for copying, distribution and modification follow. Pay close attention to the difference between a "work based on the library" and a "work that uses the library".

The former contains code derived from the library, whereas the latter must be combined with the library in order to run.

TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License Agreement applies to any software library or other program which contains a notice placed by the copyright holder or other authorized party saying it may be distributed under the terms of this Lesser General Public License (also called "this License"). Each licensee is addressed as "you".

A "library" means a collection of software functions and/or data prepared so as to be conveniently linked with application programs (which use some of those functions and data) to form executables.

The "Library", below, refers to any such software library or work which has been distributed under these terms. A "work based on the Library" means either the Library or any derivative work under copyright law: that is to say, a work containing the Library or a portion of it, either verbatim or with modifications and/or translated straightforwardly into another language. (Hereinafter, translation is included without limitation in the term "modification".)

"Source code" for a work means the preferred form of the work for making modifications to it. For a library, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the library.

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running a program using the Library is not restricted, and output from such a program is covered only if its contents constitute a work based on the Library (independent of the use of the Library in a tool for writing it). Whether that is true depends on what the Library does and what the program that uses the Library does.

1. You may copy and distribute verbatim copies of the Library's complete source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and distribute a copy of this License along with the Library.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Library or any portion of it, thus forming a work based on the Library, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

- **a)** The modified work must itself be a software library.
- **b)** You must cause the files modified to carry prominent notices stating that you changed the files and the date of any change.
- **c)** You must cause the whole of the work to be licensed at no charge to all third parties under the terms of this License.
- **d)** If a facility in the modified Library refers to a function or a table of data to be supplied by

an application program that uses the facility, other than as an argument passed when the facility is invoked, then you must make a good faith effort to ensure that, in the event an application does not supply such function or table, the facility still operates, and performs whatever part of its purpose remains meaningful.

(For example, a function in a library to compute square roots has a purpose that is entirely well-defined independent of the application. Therefore, Subsection 2d requires that any application-supplied function or table used by this function must be optional: if the application does not supply it, the square root function must still compute square roots.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Library, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Library, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Library.

In addition, mere aggregation of another work not based on the Library with the Library (or with a work based on the Library) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may opt to apply the terms of the ordinary GNU General Public License instead of this License to a given copy of the Library. To do this, you must alter all the notices that refer to this License, so that they refer to the ordinary GNU General Public License, version 2, instead of to this License. (If a newer version than version 2 of the ordinary GNU General Public License has appeared, then you can specify that version instead if you wish.) Do not make any other change in these notices.

Once this change is made in a given copy, it is irreversible for that copy, so the ordinary GNU General Public License applies to all subsequent copies and derivative works made from that copy.

This option is useful when you wish to copy part of the code of the Library into a program that is not a library.

4. You may copy and distribute the Library (or a portion or derivative of it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange.

If distribution of object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place satisfies the requirement to distribute the source code, even though third parties are not compelled to copy the source along with the object code.

5. A program that contains no derivative of any portion of the Library, but is designed to work with the Library by being compiled or linked with it, is called a "work that uses the Library". Such a

work, in isolation, is not a derivative work of the Library, and therefore falls outside the scope of this License.

However, linking a "work that uses the Library" with the Library creates an executable that is a derivative of the Library (because it contains portions of the Library), rather than a "work that uses the library". The executable is therefore covered by this License. Section 6 states terms for distribution of such executables.

When a "work that uses the Library" uses material from a header file that is part of the Library, the object code for the work may be a derivative work of the Library even though the source code is not. Whether this is true is especially significant if the work can be linked without the Library, or if the work is itself a library. The threshold for this to be true is not precisely defined by law.

If such an object file uses only numerical parameters, data structure layouts and accessors, and small macros and small inline functions (ten lines or less in length), then the use of the object file is unrestricted, regardless of whether it is legally a derivative work. (Executables containing this object code plus portions of the Library will still fall under Section 6.)

Otherwise, if the work is a derivative of the Library, you may distribute the object code for the work under the terms of Section 6. Any executables containing that work also fall under Section 6, whether or not they are linked directly with the Library itself.

6. As an exception to the Sections above, you may also combine or link a "work that uses the Library" with the Library to produce a work containing portions of the Library, and distribute that work under terms of your choice, provided that the terms permit modification of the work for the customer's own use and reverse engineering for debugging such modifications.

You must give prominent notice with each copy of the work that the Library is used in it and that the Library and its use are covered by this License. You must supply a copy of this License. If the work during execution displays copyright notices, you must include the copyright notice for the Library among them, as well as a reference directing the user to the copy of this License. Also, you must do one of these things:

- **a)** Accompany the work with the complete corresponding machine-readable source code for the Library including whatever changes were used in the work (which must be distributed under Sections 1 and 2 above); and, if the work is an executable linked with the Library, with the complete machine-readable "work that uses the Library", as object code and/or source code, so that the user can modify the Library and then relink to produce a modified executable containing the modified Library. (It is understood that the user who changes the contents of definitions files in the Library will not necessarily be able to recompile the application to use the modified definitions.)
- **b)** Use a suitable shared library mechanism for linking with the Library. A suitable mechanism is one that (1) uses at run time a copy of the library already present on the user's computer system, rather than copying library functions into the executable, and (2) will operate properly with a modified version of the library, if the user installs one, as long as the modified version is interface-compatible with the version that the work was made with.
- **c)** Accompany the work with a written offer, valid for at least three years, to give the same user the materials specified in Subsection 6a, above, for a charge no more than the cost of performing this distribution.
- **d)** If distribution of the work is made by offering access to copy from a designated place,

offer equivalent access to copy the above specified materials from the same place.
- **e)** Verify that the user has already received a copy of these materials or that you have already sent this user a copy.

For an executable, the required form of the "work that uses the Library" must include any data and utility programs needed for reproducing the executable from it. However, as a special exception, the materials to be distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

It may happen that this requirement contradicts the license restrictions of other proprietary libraries that do not normally accompany the operating system. Such a contradiction means you cannot use both them and the Library together in an executable that you distribute.

7. You may place library facilities that are a work based on the Library side-by-side in a single library together with other library facilities not covered by this License, and distribute such a combined library, provided that the separate distribution of the work based on the Library and of the other library facilities is otherwise permitted, and provided that you do these two things:

- **a)** Accompany the combined library with a copy of the same work based on the Library, uncombined with any other library facilities. This must be distributed under the terms of the Sections above.
- **b)** Give prominent notice with the combined library of the fact that part of it is a work based on the Library, and explaining where to find the accompanying uncombined form of the same work.

8. You may not copy, modify, sublicense, link with, or distribute the Library except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense, link with, or distribute the Library is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

9. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Library or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Library (or any work based on the Library), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Library or works based on it.

10. Each time you redistribute the Library (or any work based on the Library), the recipient automatically receives a license from the original licensor to copy, distribute, link with or modify the Library subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties with this License.

11. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may

not distribute the Library at all. For example, if a patent license would not permit royalty-free redistribution of the Library by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Library.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply, and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

12. If the distribution and/or use of the Library is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Library under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

13. The Free Software Foundation may publish revised and/or new versions of the Lesser General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Library specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Library does not specify a license version number, you may choose any version ever published by the Free Software Foundation.

14. If you wish to incorporate parts of the Library into other free programs whose distribution conditions are incompatible with these, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

15. BECAUSE THE LIBRARY IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE LIBRARY, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE LIBRARY "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR

PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE LIBRARY IS WITH YOU. SHOULD THE LIBRARY PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

16. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE LIBRARY AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE LIBRARY (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE LIBRARY TO OPERATE WITH ANY OTHER SOFTWARE), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Libraries

If you develop a new library, and you want it to be of the greatest possible use to the public, we recommend making it free software that everyone can redistribute and change. You can do so by permitting redistribution under these terms (or, alternatively, under the terms of the ordinary General Public License).

To apply these terms, attach the following notices to the library. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

```
one line to give the library's name and an idea of what it does.
Copyright (C) year  name of author

This library is free software; you can redistribute it and/or
modify it under the terms of the GNU Lesser General Public
License as published by the Free Software Foundation; either
version 2.1 of the License, or (at your option) any later version.

This library is distributed in the hope that it will be useful,
but WITHOUT ANY WARRANTY; without even the implied warranty of
MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the GNU
Lesser General Public License for more details.

You should have received a copy of the GNU Lesser General Public
License along with this library; if not, write to the Free Software
Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA  02111-1307  USA
```

Also add information on how to contact you by electronic and paper mail.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the library, if necessary. Here is a sample; alter the names:

```
Yoyodyne, Inc., hereby disclaims all copyright interest in
the library `Frob' (a library for tweaking knobs) written
```

```
by James Random Hacker.

signature of Ty Coon, 1 April 1990
Ty Coon, President of Vice
```